The Basics of Compiled Languages, Interpreted Languages, and Just-in-Time Compilers

f
5
(http://v
u=https
basics-
of-
compile
languag
interpre
languag
and-
just-in-
time-
compile
in-
Time%2

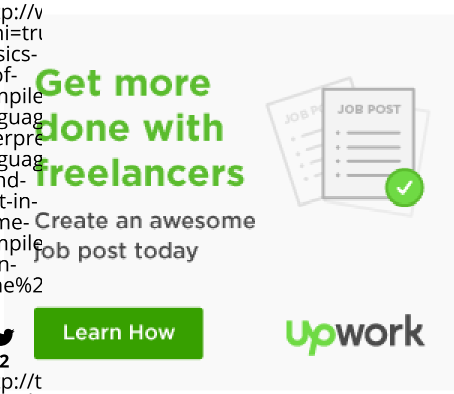
G+
1
(https://
url=http
basics-
of-
compile
languag
interpre
languag
and-
just-in-
time-
compile
in-
Time%2

in
3
(http://v
mini=tru
basics-
of-
compile
languag
interpre
languag
and-
just-in-
time-
compile
in-
Time%2

2
(http://t
text=Th
in-
Time%2
basics-
of-
compile
languag
interpre
languag
and-
just-in-
time-
compile

When we're comparing **programming languages (https://www.upwork.com/hiring/development/web-development-languages-101/)**, there are lots of ways to describe them: "high-level," front-end vs. back-end, cross-platform, and more. But one differentiator comes up often: compiled vs. interpreted. So what does it mean and why is it important?

In this article, we'll break down how code gets translated from the code *programmers write*, to the code *computers read*, the difference between compiled and interpreted code, and what makes "just-in-time" compilers so fast and efficient.


Get more done with freelancers
Create an awesome job post today
JOB POST
Learn How
upwork

HUMAN CODE VS MACHINE CODE

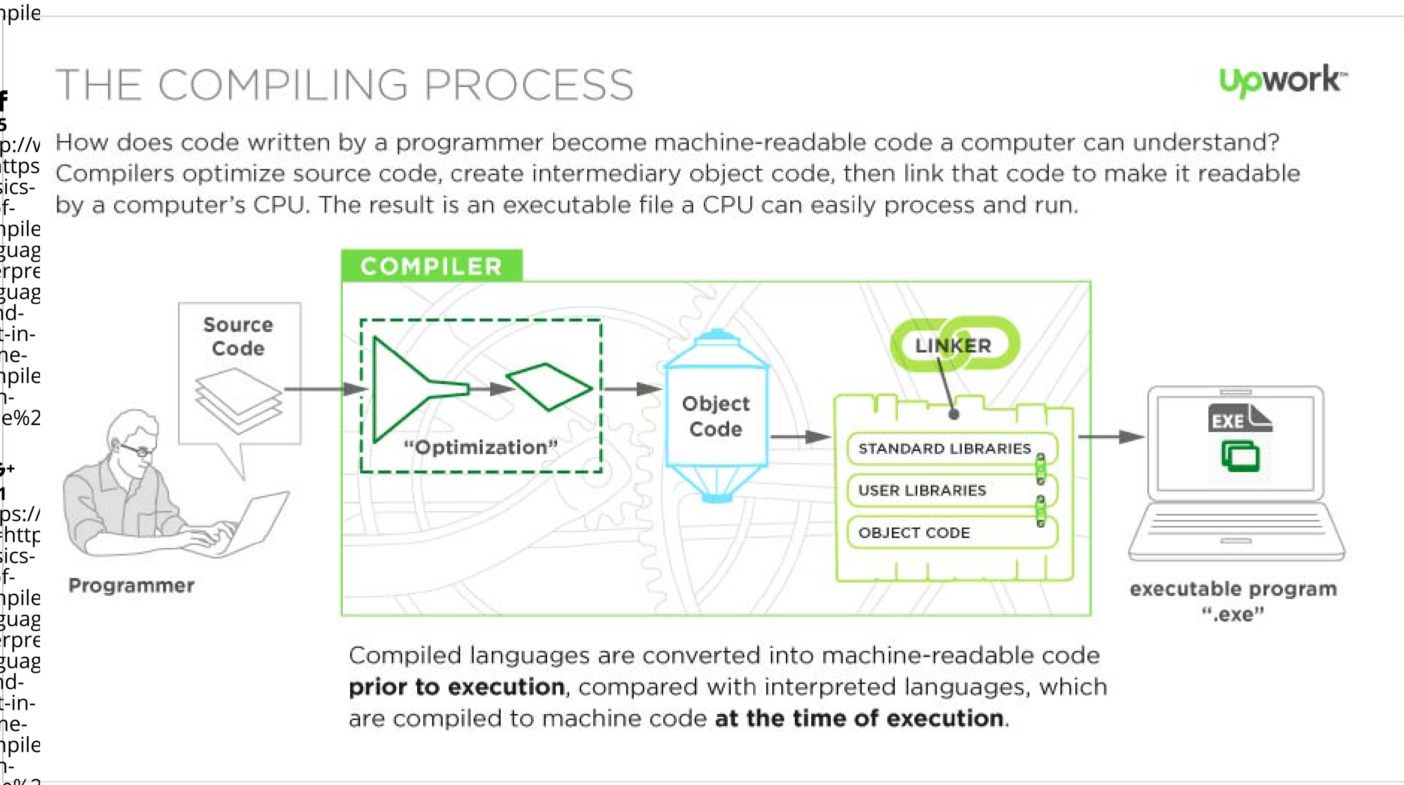
The code a programmer writes is *not* the same code a computer's operating system reads and executes—it actually has to be translated at some point before or during runtime, when the computer processes it. That original code is "**human code**," or *source code*, and the code that computers process is "**machine code**."

?

A programmer will write lines of code in the C language, for example, in a way and in an order that is logical for a human to read. But that might not always be the most logical order for a computer to process, so a translation needs to happen. The code should be in a format readable by a computer's **CPU** (central processing unit or "processor"), and compilers and interpreters make this happen. They also control how and when all of this occurs and can ultimately determine on what platforms a program can be run, and at what speed.

COMPILED LANGUAGES

are written in a code that can be executed directly on a computer's processor. This is because **a compiler** has translated the code into the computer's "native" language up front, well before the program is even run. This process can take many passes before it is optimized as machine code, but the output is always code that's ready to be executed—and that executes efficiently, as a result.



THE COMPILING PROCESS

How does code written by a programmer become machine-readable code a computer can understand? Compilers optimize source code, create intermediary object code, then link that code to make it readable by a computer's CPU. The result is an executable file a CPU can easily process and run.

Compiled languages are converted into machine-readable code **prior to execution**, compared with interpreted languages, which are compiled to machine code **at the time of execution**.

(https://content-static.upwork.com/blog/uploads/sites/3/2015/06/10132943/Compilering-process-intro.png)

A **compiler** is a program that converts human-readable code into computer-readable instructions—a process that only happens **once** in the lifespan of that code. Initially, it takes a bit longer because the compiler has to rearrange, optimize, or "compile" object code first.

Once a compiler takes in source code, optimizes it, and then generates the object code, another step has to happen. Object code can't always run on its own; it needs dynamic load libraries and other bits of code—code that's unique to the operating system, which is housed in those libraries. Within the compiling process, these libraries are linked to object code in a "linker," a part of the compiler that bundles those bits of code together, then it's good to go.

That final product—the object code packaged up with the libraries—is the compiling program's **executable file**—a .exe file.

Some compiled languages include:

- C
- **C++ (https://www.upwork.com/hiring/development/cplusplus-the-language/)**
- **C# (https://www.upwork.com/hiring/development/c-sharp-developer/)**
- Erlang
- **Objective-C (https://www.upwork.com/hiring/mobile/swift-vs-objective-c-a-look-at-ios-programming-languages/)**
- Pascal
- **Scala (https://www.upwork.com/hiring/data/scala-hybrid-functionalobject-oriented-language-big-data/)**
- **Swift (https://www.upwork.com/hiring/mobile/migrating-your-ios-app-to-swift-3/)**
- Smalltalk

7
1 (https:// view=cn in- Time%2 basics- of- compile languag interpre languag and- just-in- time- compile
f 5 (http://w u=https basics- of- compile languag interpre languag and- just-in- time- compile in- Time%2
G+ 1 (https:// url=http basics- of- compile languag interpre languag and- just-in- time- compile in- Time%2
in 3 (http://w mini=tru basics- of- compile languag interpre languag and- just-in- time- compile in Time%2
2 (http://t text=Th in- Time%2 basics- of- compile languag interpre languag and- just-in- time- compile
?

- **TypeScript (https://www.upwork.com/hiring/development/typescript-coffeescript-two-tools-to-write-faster-cleaner-javascript/)**

Recommended TypeScript Freelancers

$30/hr

Brett R.

Senior ColdFusion Developer

(/o/profiles/users/_~01cc0d4319165f0e96)

Interpreted languages

An **interpreted language** is any programming language that isn't already in "machine code" prior to runtime. Unlike compiled languages, an interpreted language's translation doesn't happen beforehand. Translation occurs *at the same time as the program is being executed.*

Many of an interpreted language's instructions can be executed directly, without compiling to machine code; however, when certain code is required, an interpreter steps in during runtime and translates it on the spot.

Interpreted languages have a major advantage: they're portable, which means they can run on different operating systems and platforms. Also, because they're translated on the spot, they're going to be optimized for the system on which they're being run. That means there are no middle steps, less memory space is required for interim object code, and there's no need to worry about platform-specific code.

HOW INTERPRETING WORKS

The **interpreter** is a program that converts source code—the human-readable code mentioned above—into machine code **each time** you run the program, one line at a time. It starts interpreting each instruction immediately upon execution, which means that the resulting program runs slower than a compiled program—it's got more going on at runtime. Compiled languages, on the other hand, have already been through this translation before program execution, so they're arguably faster.

Interpreters have some other bonuses, too. They're especially helpful for reviewing, running, and testing an application's functionality during development because they're able to execute high-level programs immediately—and generate helpful error reports. Also, they allow programmers to make small, step-by-step changes during the development process, incrementally, which complements a step-by-step process for adding and then testing smaller sections of an application.

Some interpreted languages include:

- **Java (https://www.upwork.com/hiring/development/the-java-platform/)**
- **JavaScript (https://www.upwork.com/hiring/development/what-is-javascript/)**
- **PHP (https://www.upwork.com/hiring/development/php-frameworks-hiring-a-php-developer/)**
- **Perl (https://www.upwork.com/hiring/development/the-past-present-and-future-of-the-perl-programming-language/)**
- **Python (https://www.upwork.com/hiring/development/python-programming-language/)**
- **Ruby (https://www.upwork.com/hiring/development/ruby-developer/)**

SO, TO SUMMARIZE:

7
1 (https:// view=cm in- Time%2 basics- of- compile languag interpre languag and- just-in- time- compile
f 3 (http://w u=https basics- of- compile languag interpre languag and- just-in- time- compile in- Time%2
G+ 1 (https:/ url=http basics- of- compile languag interpre languag and- just-in- time- compile in- Time%2
in 3 (http://w mini=tru basics- of- compile languag interpre languag and- just-in- time- compile in- Time%2
2 (http://t text=Th in- Time%2 basics- of- compile languag interpre languag and- just-in- time- compile
?

7

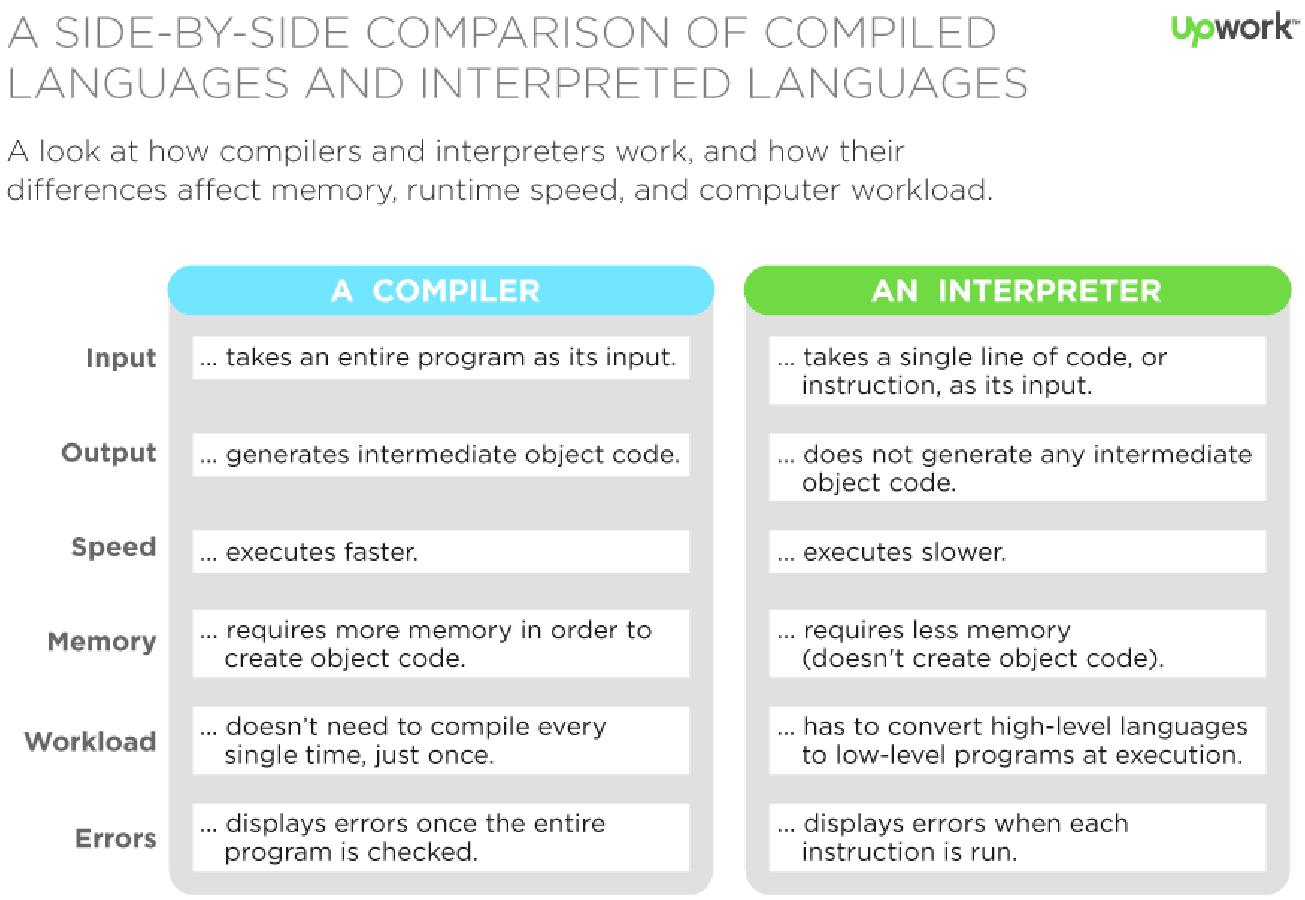
1 (https:// view=cn in- Time%2 basics- of- compile languag interpre languag and- just-in- time- compile

5 (http://v u=https basics- of- compile languag interpre languag and- just-in- time- compile in- Time%2

G+ 1 (https:// url=http basics- of- compile languag inte languag an just-in- time- compile in- Time%2

in 3 (http://v mini=tru basics- of- compile languag interpre languag and- just-in- time- compile in- Time%2

2 (http://t text=Th in- Time%2 basics- of- compile languag interpre languag and- just-in- time- compile



A SIDE-BY-SIDE COMPARISON OF COMPILED LANGUAGES AND INTERPRETED LANGUAGES

A look at how compilers and interpreters work, and how their differences affect memory, runtime speed, and computer workload.

	A COMPILER	AN INTERPRETER
Input	... takes an entire program as its input.	... takes a single line of code, or instruction, as its input.
Output	... generates intermediate object code.	... does not generate any intermediate object code.
Speed	... executes faster.	... executes slower.
Memory	... requires more memory in order to create object code.	... requires less memory (doesn't create object code).
Workload	... doesn't need to compile every single time, just once.	... has to convert high-level languages to low-level programs at execution.
Errors	... displays errors once the entire program is checked.	... displays errors when each instruction is run.

(https://content-static.upwork.com/blog/uploads/sites/3/2015/06/10132943/Compiled-vs-Interpreted-Languages-Comparison.png)

"Just in Time" (JIT) Compilers

JIT compilers are next-generation compilers, but they don't just run code—they improve it over time.

Java has a JIT compiler as part of the Java Virtual Machine (JVM) (http://www.upwork.com/hiring/development/the-java-platform/); C# has one within **the .NET framework (https://www.upwork.com/hiring/development/asp-net-framework/)**; and Android has a JIT in its Dalvik Virtual Machine (DVM). So, what are JIT compilers, and what do they do?

If typical compilers take source code and completely convert it into machine code, JITs take the same source code and convert it into an intermediary "assembly language," which can then be pulled from when it's needed. And that's the key. Assembly code is interpreted into machine code on call—resulting in a faster translation of only the code that you need.

Aside from their ability to interpret on the fly, JIT compilers are also different from standard compilers because they have access to dynamic runtime information and are able to optimize code. JITs **monitor** and **optimize** while they run. By honing in on "hot threads"—code that's getting requested more often than other code—JIT compilers are able to refine frequently used instructions and make them run better in the future.

JITs are key not only because they reduce the CPU's workload by not compiling everything all at once, but also because the resulting compiled code is optimized for *that particular* CPU. It's why languages with JIT compilers are able to be so "portable" and run on any platform or OS.

Upwork is a freelancing marketplace where businesses of all sizes can find talented professionals across multiple disciplines and categories. If you are a business and are looking to get projects done, consider signing up!

?

Join Upwork! (https://www.upwork.com/signup/?dest=home&forceHipo=1&utm_campaign=Join%20Button&utm_source=HHQ&utm_medium=content)

BY CAREY WODEHOUSE (HTTPS://WWW.UPWORK.COM/HIRING/AUTHOR/CAREYWODEHOUSE/) - FREELANCE CONTENT MARKETER AND WRITER

Carey Wodehouse is a freelance content marketer and writer based in Richmond, VA who's worked for clients ranging from online retailers and global market research... more (https://www.upwork.com/hiring/author/careywodehouse/)

RELATED SKILLS

C# (/Hire/C-Sharp-Freelancers/) C++ (Https://Www.Upwork.Com/Hiring/Tag/C-Plus-Plus/) Java (/Hire/Java-Freelancers/)

JavaScript (/Hire/Javascript-Freelancers/) PHP (/Hire/Php-Freelancers/) Python (/Hire/Python-Freelancers/)

Swift (/Hire/Swift-Freelancers/)

‹ Previous article (https://www.upwork.com/hiring/quotes/serge-k-presto-sports/) Next article ›

RELATED ARTICLES

JQUERY: THE "WRITE LESS, DO MORE" JAVASCRIPT LIBRARY (HTTPS://WWW.UPWORK.COM/HIRING/DEVELOPMENT/JQUERY-JAVASCRIPT-LIBRARY/)

October 14, 2016

jQuery is a fast, small, and feature-rich JavaScript object library that marked a significant breakthrough for front-end web development. See why you should be using jQuery on your site's front end.
Read More (https://www.upwork.com/hiring/development/jquery-javascript-library/)

IS METEORJS THE RIGHT NODE.JS FRAMEWORK FOR YOUR APP? (HTTPS://WWW.UPWORK.COM/HIRING/DEVELOPMENT/METEORJS-NODE-JS-FRAMEWORK/)

April 11, 2016

MeteorJS is one of the most talked-about Node.js frameworks on the scene. So where does it fall in the Node.js framework landscape, and is it the right choice for your MEAN-powered application?
Read More (https://www.upwork.com/hiring/development/meteorjs-node-js-framework/)

HIDDEN ADVANTAGES OF CROSS PLATFORM DEVELOPMENT WITH REACT NATIVE (HTTPS://WWW.UPWORK.COM/HIRING/FOR-CLIENTS/CROSS-PLATFORM-DEVELOPMENT-WITH-REACT-NATIVE/)

June 20, 2017

In the past choosing cross-platform development posed a number of risks, and most companies who wanted a mobile app clearly benefited from developing one for a particular mobile platform. The perception was that a cross-platform product was less user-friendly in terms of design, less reliable, and more difficult to update. Gradually, as cross-platform development tools have evolved, this perception has changed.
Read More (https://www.upwork.com/hiring/for-clients/cross-platform-development-with-react-native/)

?

7

✉ 1 (https:// view=cm in- Time%2 basics- of- compil langua interpre languag and- just-in- time- compile

f 5 (http://w u=https basics- of- compile languag interpre languag and- just-in- time- compile in- Time%2

G+ 1 (https:// url=http basics- of- compile languag interpre languag and- just-in- time- compile in- Time%2

in 3 (https:// mini=tru basics- of- compile languag interpre languag and- just-in- time- compile in- Time%2

2 (http:// text=Th in- Time%2 basics- of- compile languag interpre languag and- just-in- time- compile

6 COMMON MISTAKES FOR PHP DEVELOPERS TO AVOID (HTTPS://WWW.UPWORK.COM/HIRING/DEVELOPMENT/COMMON-MISTAKES-AVOID-PHP-PROGRAMMING/)

November 14, 2016

PHP is one of the most common languages on the web, so as a developer, it helps to have it in your tool kit. However, when learning any new language, it's possible to make some mistakes along the way. Here are six of the most common mistakes PHP developers may face and ways to help avoid them.

Read More (https://www.upwork.com/hiring/development/common-mistakes-avoid-php-programming/)

Log In (/login) | Sign Up (/signup/create-account) | Terms & Conditions (/legal/)

✉
1
(https://
view=cn
in-
Time%2
basics-
of-
compile
languag
interpre
languag
and-
just-in-
time-
compile

f
5
(http://v
u=https
basics-
of-
compile
languag
interpre
languag
and-
just-in-
time-
compile
in-
Time%2

G+
1
(https://
url=http
basics-
of-
compile
languag
interpre
languag
and-
just-in-
time-
compile
in-
Time%2

in
3
(http://v
mini=tru
basics-
of-
compile
languag
interpre
languag
and-
just-in-
time-
compile
in-
Time%2

2
(http://t
text=Th
in-
Time%2
basics-
of-
compile
languag
interpre
languag
and-
just-in-
time-
compile

?

7

✉
1
(https://
view=cn
in-
Time%2
basics-
of-
compile
languag
interpre
languag
and-
just-in-
time-
compile